

基于 Contig 的单面基因组片段填充问题研究

柳楠*, 朱永琦, 李胜华, 崔晓宇

(山东建筑大学 计算机科学与技术学院, 山东 济南 250101)

摘要:近些年来,随着基因测序技术的继续发展与应用,大量不完整基因组片段的处理问题有待研究。同时由于目前大部分的生物学研究是基于基因组序列可以提供完整信息的假设,但通过生物测序技术获得一个完整的基因组序列仍是困难的。因此基因组重组问题在计算生物学领域愈发受到关注和研究,研究如何填充缺失基因组使其完整,具有重要意义。针对单面基因组片段填充算法,目前常采用最大化公共邻接数目的度量依据,是将缺失基因填充至不完整基因序列中得到填充后的重排列基因序列,使之与参照基因序列之间的新公共邻接数目最大。主要研究了基于 contig(片段重叠群)的单面重复基因组填充问题,重点对该问题的现有算法在近似比、核心技术以及时间复杂度等多方面进行了对比分析与总结,并分别提出了各类算法的改进思路,有助于进一步研究基于 contig 的单面序列填充问题。

关键词:计算生物学;基因组;片段填充;近似算法;NP-完全

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2022)11-0008-08

doi:10.3969/j.issn.1673-629X.2022.11.002

Research Progress of One-sided Repetitive Genome Scaffold Filling Based on Contig

LIU Nan*, ZHU Yong-qi, LI Sheng-hua, CUI Xiao-yu

(School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China)

Abstract: In recent years, with the continuous development and application of gene sequencing technology, the number of incomplete genome scaffolds needs to be studied. At the same time, most of the current biological research is based on the assumption that genome sequences can provide complete information, but it is still difficult to obtain a complete genome sequence by biological sequencing technology. Therefore, genome recombination has attracted more and more attention and research in the field of computational biology. It is of great significance to study how to fill the missing genome and make it complete. The one-sided genome scaffold filling algorithm is to fill the missing genes into the incomplete genome scaffold to obtain the filled rearranged genome scaffold, and maximizes the number of common adjacencies between it and the reference genome scaffold. We mainly study the one-sided repeated genome scaffold filling problem based on contig, analyze and summarize the existing algorithms and their time complexity. We focus on the comparative analysis and summary of the existing algorithms in approximation ratio, core technology and time complexity, and put forward the improvement ideas of various algorithms, which is helpful for the further study of one-sided scaffold filling problem based on contig in the future.

Key words: computational biology; genome; scaffold filling; approximation algorithms; NP-complete

0 引言

随着二十世纪三大科学计划之一的人类基因组计划的实施,大量的生物学数据有待处理^[1-4],如何利用计算机建模、仿真等技术去提取其中有用的数据,进而研究其中所蕴含的生物学意义,对计算机科学技术来说是一项严峻的挑战^[5-6]。因此在二十世纪提出了一门新兴交叉学科—计算生物学。计算生物学运用数学、计算机和生物学相关理论解决生物学问题,已经成

为目前最活跃的研究领域之一^[7-9]。

基因组片段填充问题^[10-12]是计算生物学极其经典的问题之一,其中含重复基因的基因组片段填充问题已经被证明为 NP-完全问题^[13-14],如何优化基因组片段填充近似算法是近些年来的讨论热点。依据基因样本序列中是否含有重复基因,将该基因组填充问题分为含重复基因的基因组片段填充问题和无重复基因的基因组片段填充问题;或依据基因样本序列不完整

收稿日期:2021-11-07

修回日期:2022-03-10

基金项目:国家自然科学基金(61902221);山东省自然科学基金(ZR2018MF012)

作者简介:柳楠(1980-),女(满族),博士,副教授,CCF会员(H4774M),通讯作者,研究方向为算法分析与设计、复杂性理论和生物计算;
朱永琦(1999-),女,硕士研究生,研究方向为算法分析与设计、计算基因组学。

数量,将该基因组填充问题分为单面基因组片段填充和双面基因组片段填充,其中一条序列完整,另一条序列缺失,称为单面基因组序列,两条基因序列均为不完整的,则为双面基因组序列^[15]。

该文重点讨论单面重复基因组片段填充问题。Munoz 和 D. Sankoff 等人^[12-13]首次提出了基于最小重组距离(DCJ 距离)的单面基因组填充方法,使用断点图设计了多项式时间算法,并证明了基于 DCJ 距离的单面基因组片段填充算法是多项式可解的。对于单面无重复基因组片段填充问题,H. Jiang 等人提出了使用 DCJ 距离或断点距离为度量的算法,并证明了其是多项式可解的^[14];对于含重复基因的基因组片段填充问题,H. Jiang 等人证明了其是 NP-完全的,并提出了 4/3-近似算法^[14-16]。随后 N. Liu 等人采用局部优化和贪婪算法将该类问题近似度改善到 1.25^[17-18];J. Ma 等人采用非盲局部搜索策略将该类问题近似度进一步改善到 1.2^[19-20]。

在许多应用中,基因组序列通常被定义为一连串连续的片段重叠群(contig)^[21],其中任何一个 contig 都不能被破坏,缺失基因的插入只能在 contig 的两端执行。在此约束下,当不存在重复基因时,单面基因组片段填充问题是多项式可解的;当存在重复基因时,H. Jiang 等人通过最大化公共邻接证明了该类问题是 NP-完全的,并提出了一个近似值为 2 的近似算法^[22-23]和一个双参数的 FPT 算法^[22](k , 公共邻接数, d , 基因最大重复数);L. Bulteau 等人给出了一种基于最大邻接数和最小断点距离的 k -Mer 参数的 FPT 算法^[24];Q. Feng 等人通过构造辅助图和二次寻找最大匹配给出了 2.57-近似算法^[25]。

该文的主要工作有以下三个方面:系统归纳了基于 contig 的单面基因组片段填充问题的现有算法并通过实例实现了算法,有助于读者对此类问题的进一步了解;在技术应用和时间复杂度等方面对现有算法做了对比,并分析了这些算法存在的一些弊端;分析接下来研究工作中面临的挑战和可能的解决方案。

1 相关定义

该文只关注基于 contig 的单面基因组片段填充算法,但其结果可以推广到多染色体或环状基因组。

首先,给出一些必要的定义。不失一般性,假设所有的基因和基因组都由无符号的字母和整数组成,给定一个集合 Σ 和一个基因序列 S ,使用 $c(S)$ 表示基因序列 S 中所有符号的集合。如果 Σ 中的符号在基因序列 S 中出现且只出现一次,则称 S 是 Σ 上的一个排列,否则称为序列。对于 Σ 中的任意两个符号 x, y ,如果基因序列 S 至少包含 $\{xy, yx\}$ 中的任意一个子集,那

么则称 x, y 在 S 中邻接,令 $P(S)$ 为 S 中所有邻接的集合。设 A 和 B 是 Σ 中的两个基因序列, $A = \{a_1 a_2 \cdots a_n\}$, $B = \{b_1 b_2 \cdots b_m\}$ 。对于 $P(A)$ 中的任意一个邻接 $a_i a_{i+1}$ 和 $P(B)$ 中的任意一个邻接 $b_j b_{j+1}$,如果 $a_i a_{i+1} = b_j b_{j+1}$ (或 $a_i a_{i+1} = b_{j+1} b_j$),则称 $a_i a_{i+1}$ 与 $b_j b_{j+1}$ 构成了公共邻接, $a(A, B)$ 表示 A 和 B 的公共邻接集合,同时称 $(a_i a_{i+1}, b_j b_{j+1})$ 为一个匹配对。如果 $P(A)$ 和 $P(B)$ 中不存在 $a_i a_{i+1} = b_j b_{j+1}$ (或 $a_i a_{i+1} = b_{j+1} b_j$),则称 $a_i a_{i+1}$ 相对于 $b_j b_{j+1}$ 构成了断点, $bp(A, B)$ 和 $bp(B, A)$ 分别表示 A 和 B 的断点集合,如图 1 所示。

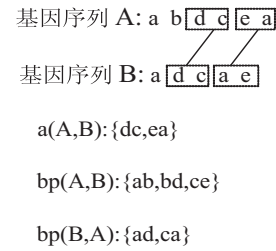


图 1 邻接、断点的示例

定义一个基因组序列是由一系列 contig 构成的,且 contig 内部不能插入缺失基因,即 $S = \langle C_1, C_2, \dots, C_m \rangle$, 其中 C_i 为一个片段重叠群。

下面具体给出 One-Sided-SF-max 问题的概念:

定义 1: One-Sided-SF-max 问题。

输入: 一个完整的基因组序列 G 和一个不完整的基因组序列 S , 其中 $S = \langle C_1, C_2, \dots, C_m \rangle$, 基因组序列 G 和片段重叠群 C_i 中的基因元素均来自于符号集合 Σ , 且缺失基因集合 $X = c(G) - c(S) \neq \emptyset$ 。

输出: 将 $X = c(G) - c(S) \neq \emptyset$ 插入 S 得到 S' , 使得 $|a(S', G)|$ 最大。

2 One-Sided-SF-max 问题

One-Sided-SF-max 问题已经被证明为 NP-完全的^[22], 此类问题不能在有效时间内求出精确解, 因此设计近似算法更具有实际意义。本节主要对 One-Sided-SF-max 问题进行简要介绍, 概括分析了国内外经典的三种算法: 2-近似算法、2.57-近似算法以及 k -Mer 算法。

2.1 One-Sided-SF-max 问题的 2-近似算法

该算法由 H. Jiang 等人提出, 主要使用了贪婪和最大匹配的思想来实现基于 contig 的单面基因组片段填充。首先在该算法中给出以下定义: 对于基因序列 $S = \langle C_1, C_2, \dots, C_m \rangle$, 定义 α_i 和 β_i 分别是 contig C_i 的首尾元素, 其中 $i \in [1, m]$ 。 $\langle \beta_i, \alpha_{i+1} \rangle$ 构成一个 slot, 缺失基因只能插入到 slot 中。在 S 的两端有两个开放 slot, 分别表示为 $\langle -\infty, \alpha_1 \rangle$ 和 $\langle \beta_m, +\infty \rangle$ 。对于缺失基因 x , 如果存在一个公共邻接 xy (或 yx),

其中 $y = \alpha_i$ 或 $y = \beta_i$, 则称公共邻接 xy (或 yx) 为外部邻接, 否则称公共邻接 xy (或 yx) 为内部邻接。

定义缺失基因集合 X 中有一个长度为 n 的子串, 如果插入到 $\text{slot} < \beta_i, \alpha_{i+1} >$ 中 ($1 \leq i \leq m-1$), 产生 $n+1$ 个新公共邻接, 称子串为 n -Type-1 类型串。同样的, 产生 n 个新公共邻接, 称为 n -Type-2 类型串; 产生 $n-1$ 个新公共邻接, 称为 n -Type-3 类型串。算法大体流程如下:

(1) 计算缺失基因集合 $X = c(G) - c(S)$;

(2) 对于缺失基因集合, 采用贪婪策略将 1-Type-1 类型串插入到相应的 slot 中, 并将该 slot 锁定, 不允许其他缺失基因插入;

(3) 构造二分图并求其最大匹配, 将 1-Type-2 类型串插入到可构成外部邻接的 slot 处, 并对该 slot 进行更新: 如果 x_j 插入到 $\text{slot} \bullet a_i$ 前面, 那么将此 slot 更新为 $\bullet x_j$, 如果 $\bullet x_j$ 插入到 $\text{slot} \beta_i \bullet$ 后面, 那么将此 slot 更新为 $x_j \bullet$;

(4) 以步骤 2 后的缺失基因为顶点构造多重图: 若 $x \in X, y \in X$ 且 xy 为 G 中一个内部邻接, 那么 x 和 y 之间添加一条边, 寻找最大匹配 M 。对于最大匹配 M 中的所有匹配对 xy , 如果 x 为步骤 3 中插入的元素, 则将 y 插入到相应 slot 处使得 xy 构成邻接; 将其余匹配对 xy 任意插入到未锁定 slot 中, 且不能破坏现有邻接;

(5) 在不破坏现有邻接关系的前提下, 将所有剩余缺失基因任意插入到 S 中未锁定的 slot 处;

(6) 得到近似解 S' 。

对于该算法, 下面通过一个实例 (如图 2 所示) 来说明算法的执行过程:

基因序列 $G = \langle 1, a, c, b, 1, d, 7, 5, 2, 4, g, a, 2, k, 7, d, 2 \rangle$

基因序列 $S = \langle \boxed{4, k, 1}, \boxed{c, b}, \boxed{1} \rangle$

缺失基因集合 $X = \langle a, d, 7, 5, 2, g, a, 2, 7, d, 2 \rangle$

图 2 算法实例

(1) 使用贪婪策略搜寻 1-Type-1 类型串, 找到 a 为 1-Type-1 类型串, 将 a 插入到 $\boxed{4, k, 1}$ 和 $\boxed{c, b}$ 之间, 并将此 slot 锁定, 不允许其他缺失基因插入。

(2) 搜寻 1-Type-2 类型串, 找到 $2, d, g$ 为 1-Type-2 类型串, 并以 1-Type-2 类型串集合和未锁定 slot 集合为顶点, 构造二分图, 建立的二分图 BG_1 如图 3 所示。

求得二分图中最大匹配, 将 g 插入到 $\boxed{4, k, 1}$ 前, 将 d 插入到 $\boxed{1}$ 之后。

(3) 以步骤 2 之后的剩余缺失基因集合为顶点构造多重图 Q , 建立的多重图 Q 并求得最大匹配, 将 a 插入到 g 之前, 将 7 插入到 d 之后, 将 52 插入到 a

之前。

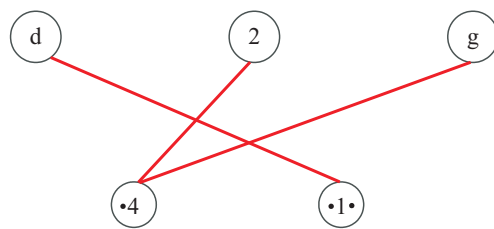


图 3 二分图 BG_1

(4) 在不破坏现有邻接的基础上, 将所有剩余缺失基因插入到未锁定的 slot 中, 此例中, 将 2 和 7 依次插入到 7 (步骤 3 中插入) 之后, 将 d 和 2 依次插入到 $\boxed{c, b}$ 和 $\boxed{1}$ 之间。

(5) 算法结束, 得到填充后的基因序列 $S' = \langle 52ag4k1acbd21d727 \rangle$ 。

以上可以看出, 通过此算法可以得到 8 个新公共邻接, 同时此例的最优解为 $S^* = \langle 2ag4k1acb1d7527d2 \rangle$, 有 13 个新公共邻接。

设 b_{ij} 表示 OPT 中 i -Type- j 类型串的个数, B_{ij} 是 OPT 中 i -Type- j 类型串的对应该集, 则 $\text{OPT} = \sum_{i \geq 1} (i+1)b_{i1} + \sum_{i \geq 1} ib_{i2} + \sum_{i \geq 1} (i-1)b_{i3}$ 。设 b'_{ij} 表示近似解中 i -Type- j 类型串的个数, B'_{ij} 是近似解中 i -Type- j 类型串的对应该集, $Y_{i,j}$ 是通过近似算法将某些最优解中的 i -Type- j 类型串转换为 Type- j 的集合。

由于步骤 2 使用了贪婪策略, 导致最优解中的一些类型串被破坏, 其中最为关键的是 B'_{11} (与 B_{11} 相比) 中错放的 1-Type-1 类型串最多可以将最优解中的一个 1-Type-1 类型串变成 Type-3 类型串, 或将两个 Type-1 子串 (v -Type-1 类型串和 w -Type-1 类型串 ($v, w \geq 1$)) 分别变成 Type-2 类型串。利用二分图最大匹配, $Y_{i,2}$ 中的每个子串生成一个外部邻接, 由此步骤 2 会产生 $b'_{12} + \sum_{i \geq 1} |Y_{i,2}|$ 个新公共邻接。

该算法近似值 $\text{APP} = (2b'_{11} + \sum_{i \geq 1} |Y_{i,2}|) + |M|$, 其中 $|M|$ 为步骤 4 中获得的最大匹配。因为步骤 2 使用的贪婪策略使得部分 Type-1 类型串被破坏, 则有:

$$|M| \geq \frac{1}{2} \left\{ \sum_{i=2 \cdots p} (i+1)b_{i1} + \sum_{i=2 \cdots q} ib_{i2} + \sum_{i=2 \cdots r} (i-1)b_{i3} \right\} + \left(\sum_{i \geq 2} \left\lfloor \frac{i}{2} \right\rfloor |Y_{i,2}| + \sum_{i \geq 2} \left\lfloor \frac{i}{2} \right\rfloor |Y_{i,3}| \right)$$

所以,

$$\text{APP} \geq 2b'_{11} + (b'_{12} + \sum_{i \geq 1} |Y_{i,2}|) + \frac{1}{2} \left\{ \sum_{i=2 \cdots p} (i+1)b_{i1} + \sum_{i=2 \cdots q} ib_{i2} + \sum_{i=2 \cdots r} (i-1)b_{i3} \right\} +$$

$$(\sum_{i \geq 2} \lfloor \frac{i}{2} \rfloor |Y_{i,2}| + \sum_{i \geq 2} \lfloor \frac{i}{2} \rfloor |Y_{i,3}|)$$

通过步骤 2,可以保证该算法能够为所有 Type-2 类型串生成一个外部邻接,通过步骤 3,可以保证对于每一个 n -Type-2 和 n -Type-3 类型串,至少可以生成 $\lfloor \frac{n}{2} \rfloor$ 个内部邻接。因此,在步骤 3 中生成的邻接数满足:

$$2b_{11} + \sum_{i \geq 1} (1 + \lfloor \frac{i}{2} \rfloor |Y_{i,2}|) +$$

$$\sum_{i \geq 2} \lfloor \frac{i}{2} \rfloor |Y_{i,3}| \geq b_{11}$$

所以有近似解:

$$\text{APP} \geq b_{11} + b_{12} + \frac{1}{2} \{ \sum_{i=2 \dots p} (i+1)b_{i1} + \sum_{i=2 \dots r} (i-1)b_{i3} \} \geq \frac{1}{2} \text{Opt}$$

该算法为一个 2-近似算法,同时该算法的运行时间主要由步骤 2 中计算 $O(n)$ 个顶点的二分图中的最大匹配以及步骤 3 中计算 $O(n)$ 个顶点的多重图中的最大匹配决定,两者都需要 $O(n^{2.5})$ 个时间,所以 2-近似算法的时间复杂度为 $O(n^{2.5})$ 。

2.2 One-Sided-SF-max 问题的 2.57-近似算法

Q. Feng 提出的 2.57-近似算法继续考虑了冗余块对填充过程存在的影响。该算法主要使用了最大匹配算法构造简单路径来具体实现基于 contig 的单面基因组片段填充问题。

首先在该算法中给出以下定义:令 $F(S)$ 为 contig C_i 的首尾元素集合, $F(S) = (\alpha_1, \beta_1, \dots, \alpha_m, \beta_m)$ 。如果最大匹配 M 中有块 xy , xy 在最大匹配 M 中出现的次数称为 xy 的指示数。设 xy 与 $bp(G, S)$ 中块 ab 可构成匹配对 (xy, ab) , 若 xy 在 M 中出现次数大于 ab 在 $bp(G, S)$ 中出现次数,则称块 xy 为一个冗余块。定义 \oplus 为对称差, $A \oplus B = (A \setminus B) \cup (B \setminus A)$, 令 K 为此算法中的两次最大匹配 M_1 和 M_2 的对称差,则 K 中每个连通分量必为简单路径或简单循环。算法的大体流程如下:

(1) 计算缺失基因集合 $X = c(G) - c(S)$ 和断点集合 $bp(G, S)$;

(2) 基于缺失基因集合 X 、断点集合 $bp(G, S)$ 和 S 中每个 contig 首尾元素集合 $F(S)$ 构造一般图 Γ_1 , 寻找最大匹配 M_1 ;

(3) 对于 M_1 中的任意块 xy , 有以下三种情况:如果 x 和 y 分别为同一个 slot 的前后两端,则合并此相邻的两个 contig;如果 x (或 y) 属于 $F(S)$, 将 y (或 x) 插入相应的 slot 中使得 xy 邻接;如果 x 和 y 均不属于 $F(S)$, 则将其置于图 H 中顶点。依据以上更新基因

序列为 S_1 ;

(4) 基于 G 、 S_1 和 H , 求得断点集合 $bp(G, S_1)$ 和 $F(S_1)$;

(5) 更新图 H : 删除可与 $bp(G, S_1)$ 中断点构成匹配对的边;

(6) 基于缺失基因集合 X 、断点集合 $bp(G, S_1)$ 和集合 $F(S_1)$ 构造一般图 Γ_2 , 寻找最大匹配 M_2 ;

(7) $\Delta = H \oplus M_2$;

(8) 对于图 Δ 中的任意路径 $k = p_1 p_2 \dots p_{l-1} p_l$, 判断其是否为简单路径: 若为简单路径, 插入到相应 slot 中, 反之删除路径 k 中任意一条边得到新的路径 $p_1 p_2 \dots p_{l-1} p_l$, 将路径 $p_1 p_2 \dots p_{l-1} p_l$ 插入到基因序列的最右侧; 更新基因序列为 S_2 ;

(9) 统一将 $c(G) - c(S_2)$ 插入到序列 S_2 的最右侧;

(10) 得到填充完成后的基因序列 S' 。

下面通过上述 2-近似算法的同一个实例(见图 2)来说明算法的执行过程:

(1) 计算断点集合 $bp(G, S) = \langle 1a, ac, 1d, d7, 75, 52, 24, 4g, ga, a2, 2k, k7, 7d, d2 \rangle$, 计算 S 中每个 contig 首尾元素集合 $F(S) = \langle 4, 1, c, b, 1, 1 \rangle$;

(2) 构造图 Γ_1 : $X \cup F(S)$ 中的所有元素被视为顶点, 对于其中任意两个元素 x, y , 如果有 $x \in X, y \in X$ 或 $y \in F(S)$ 且存在一个断点 β 使得与 xy 构成一个匹配对, 则在 x 与 y 之间添加一条边; 如果有 $x, y \in F(S)$, 假设 x 在 contig C_1 中且为 C_1 中最后一个元素, y 在 contig C_2 中且为 C_2 中第一个元素, C_1 和 C_2 相邻且存在一个断点 β 使得与 xy 构成一个匹配对, 则在 x 与 y 之间添加一条边。在图 Γ_1 中寻找最大匹配 M_1 , 如图 4 所示;

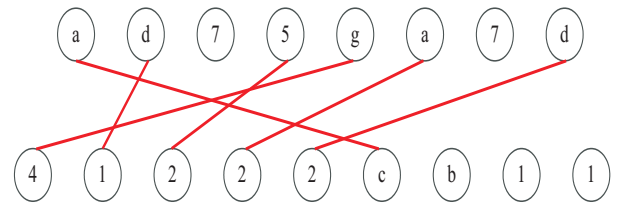


图 4 最大匹配 M_1

(3) 删除 M_1 中的冗余块: 对于 M_1 中的块 xy , xy 与断点 ω 可构成匹配对 (xy, ω) , 如果 xy 在 M_1 中的出现次数大于 ω 在 $bp(G, S)$ 中的出现次数, 则称块 xy 为冗余块并将其删除;

(4) 令 $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$ 为 M_1 中删除冗余块后剩余块的集合, H 为 $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$ 中的块与 $bp(G, S)$ 中的断点构成匹配对的集合, 则此例中 $H = \{(ac \leftrightarrow ac), (d1 \leftrightarrow 1d), (52 \leftrightarrow 52), (g4 \leftrightarrow 4g), (a2 \leftrightarrow a2), (d2 \leftrightarrow d2)\}$;

(5) 基于 H' 更新 S : 此例中, 将 g 插入到 $[4, k, 1]$ 之前, 将 a 插入到 $[c, b]$ 之前, 将 d 插入到 $[1]$ 之前, 得到新的基因序列 S_1 , 同时将块 $ac, d1, g4$ 在 H' 中删除;

(6) 更新 H' 和 S_1 : $H' = \{(52 \leftrightarrow 52), (a2 \leftrightarrow a2), (d2 \leftrightarrow d2)\}$, $S_1 = \langle \cdot [g, 4, k, 1] \cdot [a, c, b] \cdot [d, 1] \cdot \rangle$;

(7) 计算 $X' = c(G) - c(S_1)$ 得到缺失基因集合 $X' = \langle 7, 5, 2, a, 2, 7, d, 2 \rangle$, 计算 S_1 中每个 contig 首尾元素集合 $F(S_1) = \langle g, 1, a, b, d, 1 \rangle$;

(8) 计算新的断点集合 $bp(G, S_1) = \langle b1, d7, 75, 52, 24, ga, a2, 2k, k7, 7d, d2 \rangle$;

(9) 使用构造图 Γ_1 的同样方法构造图 Γ_2 ;

(10) 如果图 Γ_2 与图 H' 中存在相同边, 则在 Γ_2 中删除此条边, 此例中, 无此类边;

(11) 在图 Γ_2 中求得最大匹配 M_2 , 如图 5 所示。

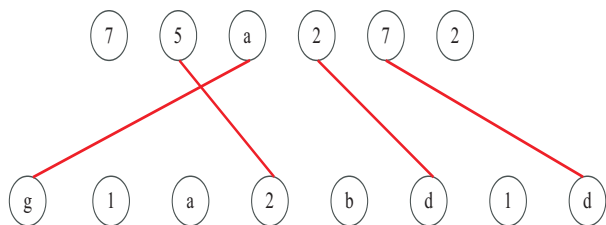


图 5 最大匹配 M_2

(12) 令 $\Delta = H' \oplus M_2$, 此例中 $\Delta = (a2, ag, 7d)$;

(13) 对于图 Δ 中的任意路径 $k = p_1 p_2 \cdots p_{i-1} p_i$, 判断其是否为简单路径: 若为简单路径, 则依据路径的首尾顶点 p_1 和 p_i 是否属于 $F(S_1)$: 如果同属于 $F(S_1)$ 但不属于同一个 slot, 那么将 $p_2 \cdots p_{i-1}$ 插入到含有 p_1 的 contig 中, 使得 p_1 与 p_2 邻接; 如果同属于 $F(S_1)$ 且同属于一个 slot, 那么将 $p_2 \cdots p_{i-1}$ 插入到此 slot 中, 使得 p_1 与 p_2 邻接, p_{i-1} 与 p_i 邻接; 反之删除路径 k 中任意一条边得到新的路径 $p_1 p_2 \cdots p_{i-1} p_i$, 将路径 $p_1 p_2 \cdots p_{i-1} p_i$ 插入到序列 S_1 的最右侧; 此例中, 将 a 插入到 $[g, k, 4, 1]$ 的之前, 将 $2, 7, d$ 依次插入到 $[d, 1]$ 之后;

(14) 更新 S_1 为 S_2 : $S_2 = \langle \cdot [a, g, 4, k, 1] \cdot [a, c, b] \cdot [d, 1, 2, 7d] \cdot \rangle$;

(15) 将剩余缺失基因插入到序列 S_2 的最右侧;

(16) 得到填充完成后的基因序列 $S' = \langle ag4k1acbd127d5722 \rangle$ 。

该算法可以得到 8 个公共邻接, 同时其中一个最优解为 $\langle 2ag4k1acbd1d7527d2 \rangle$, 有 13 个新公共邻接。

令 A_1 为第一次最大匹配 M_1 获得的最大匹配块集合, 则 A_1 的长度至少为 $\frac{|Opt|}{3}$, 令 A_2 为第二次最大匹

配 M_2 获得的最大匹配块集合, 则 A_2 的长度至少为 $\frac{|Opt|}{18}$, 设 A 为该算法的近似解, 由于无冗余块集合为 $A_1 \cup A_2$, 因此,

$$|A| = |A_1| + |A_2| \geq \frac{|Opt|}{3} + \frac{|Opt|}{18} = \frac{7}{18} |Opt|$$

该算法的近似性能比为 2.57。

2.3 One-Sided-SF-max 问题的 k-Mer 近似算法

k-Mer 算法从参数化复杂性的角度研究基因组填充问题, 相较于 H. Jiang 等人提出的 2-近似算法, 主要有以下三个方面的不同:

(1) 不再限制插入的基因集合为 $c(G) - c(S)$, 插入集合可以包含比 $c(G) - c(S)$ 更多或更少的基因集合;

(2) 允许将要插入的字符串数量预先指定为输入约束, t_1 为要插入的字符串数量的下限, t_2 为要插入的字符串数量的上限 ($t_1 \leq t_2$);

(3) 作为相似性度量依据, 不局限于最大化公共邻接的数量, 相反, 对于一个预定的参数 k , 最大化公共 k-mers 的数目, k 值越高, 结果越准确。

L. Bulteau 等人对于此类问题给出以下定义: 对于两个基因序列 G 和 S , $G \circ S$ 表示二者的串联。存在一个正整数 k , 使得 $a_k(G) = \{S[i, i+k] \mid i \in [n-k]\}$ 为序列 G 中 k-mers 的集合, 则 $a_k(G, S) = a_k(G) \cap a_k(S)$ 。设 $S[i]$ 表示 S 中第 i 个元素, $S[i, j]$ 表示序列 S 中从位置 i 到 j 的基因元素。对于一个完整基因序列 G 和一个不完整基因序列 S , 令 $p_k(S, G) = a_k(G) \setminus a_k(S)$ 表示存在于 G 中但不存在于 S 中的 k-mers 集合, 并将此类 k-mers 称为潜在的公共 k-mers。

定义 2: k-Mer Scaffold Filling (k-Mer-SF)。

输入: 一个完整的基因组序列 G 和一个不完整的基因组序列 S , 其中 $S = \langle C_1, C_2, \dots, C_m \rangle$, 且存在一个字符集合 T 和两个整数 t_1, t_2 , 有 $t_1 \leq t_2 \leq |T|$ 。

输出: 找到 $T' \subseteq T$, $t_1 \leq |T'|$ 且填充后的 $S' \in S + T'$, 使得 $|a_k(S', G)|$ 最大。

该文给出一个 k-Mer-SF 实例, 在此只举例说明了 $k=2$ 和 $k=3$ 的填充情况 (如图 6 所示)。

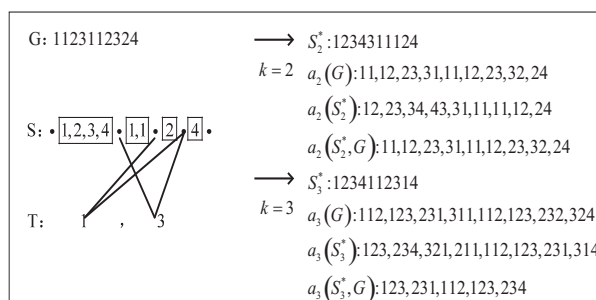


图 6 k-Mer-SF 实例

H. Jiang 等人提出的 2-近似算法和 Q. Feng 等人提出的 2.57-近似算法均为 k-Mer-SF 中 $t_1 = t_2 = |T|$ 且 $k = 2$ 时的特殊情况,并可在多项式时间 $O(\ell^3 \cdot (\ell + m)^2)$ 内计算。k-Mer-SF 相较于以上两个近似算法,不再仅仅参考公共邻接数目的多少,主要使用以下评估参数: k , k-mers 的长度; ℓ : $= a_k(S^*, G) - a_k(S, G)$, 匹配后带来的额外公共 k-mers 数目; d , 一个基因在 G 中出现的最大次数; m , S 中重叠群的个数; t_2 , 要插入的字符串数量的上限; λ , T 中字符串长度的上界。

k-Mer 算法主要解决了基于动态规划如何在 $2^{O(\ell)} \cdot n^{O(\ell)}$ 时间内求得近似解并给出其参数为 $k + \ell$ 的 FPT 算法。首先使用着色法对 k-mers 进行分类, $\beta: T \rightarrow [t_2]$ 表示潜在的公共 k-mers, $\beta: T \rightarrow [t_2]$ 表示可能插入字符。图着色后,使用动态规划算法重建序列 S , 使得 S 中有 ℓ 个潜在的公共 k-mers 转为已实现的 k-mers, 在动态规划过程中逐步找到大小递增的局部最优解, 从左到右依次将缺失基因插入到序列 S 中, 并使用局部优化策略避免一些缺失基因重复插入。

对于 k-Mer-SF 问题, 首先对序列 G , S 及插入基因集合 T 做约简操作。删除 T 中多余基因元素, 如果 T 中存在一个基因元素出现次数大于 t_2 , 将删除一个元素; 其次, 对 G 中的邻接关系分类, 并只保留潜在公共邻接, 假设 x 为不出现在 S 和 T 中的基因, y 为不出现在 G 中的基因, 令 $P_1 \circ x \circ P_2 \circ \dots \circ P_{q-1} \circ x \circ P_q$ 替换 G 且用 $C_i[1] \circ y \circ C_i[|C_i|]$ 表示 S 。如果有一个潜在邻接在 G 中出现 ℓ 次, 则在 G 中删除一个该邻接, 删除后若邻接满足以下条件之一, 则称为可实现邻接:

- (1) $b \in T$ 且 $c \in T$;
- (2) 存在一个 contig C_i 使得 $b \in C_i[|C_i|]$ 且 $c \in T$;
- (3) 存在一个 contig C_i 使得 $b \in C_i[|C_i|]$ 和 $c \in C_i[1]$;
- (4) 存在一个 contig C_i 使得 $b \in T$ 且 $c \in C_i[1]$ 。

建立邻接图 $H = (V, E)$: 令 T , G 和 S 中的基因元素作为 H 中的顶点, 如果邻接 bc 或邻接 cb 均为可实现邻接, 则令两个顶点 b 和 c 相邻, 求得最大匹配 M 。

令 $V(M)$ 表示匹配的端点, 建立两个二分图 H^1 和 H^2 且顶点分别为 $B := V(M)$ 和 $C := (V \setminus V(M))$ 。在 H^1 中, 当 bc 是一个可实现邻接时, 在 $b \in B$ 和 $c \in C$ 之间添加一条边。在 H^2 中, 当 cb 是一个可实现邻接时, 在 $b \in B$ 和 $c \in C$ 之间添加一条边。如果 H^1 中存在顶点 $b \in B$ 且度数至少为 $2\ell + m + 1$, 则将邻接 bc 从 G 中移除; 如果 H^2 中存在顶点 $b \in B$ 且度数至少为 $2\ell + m + 1$, 则将邻接 bc 从 G 中移除, 其中 c 是 b 的任意邻接。

完成约简操作后, 这些顶点的数量最多为 $|V(M)| \cdot 2 \cdot (2\ell + m + 1)$ 。这给出了 G 中顶点数量的界限, 从而给出了实例大小的界限且所有约简规则可以在多项式时间内执行。

从更广泛的角度来看, k-Mer-SF 考虑了字符串的基本插入问题。事实上, 可以将该算法扩展到更一般的情况, 即给定一个字符串 G 和一个部分字符串 S , 完成部分字符串 S 的插入得到新的字符串 S' , 使得 G 和 S' 的相似度最优, 因此其包含了 H. Jiang 等人的问题作为特例。

3 One-Sided-SF-max 问题的总结

该文发现 2-近似算法没有考虑断点对填充过程的影响和长度大于 1 的缺失串的插入情况, 2.57-近似算法则没有考虑 n -Type-3 类型串的插入情况, k-Mer 算法也仅仅介绍了固定基因子串长度的一般处理情况, 然而不同长度以及不同类型串的插入都会对公共邻接数造成影响, 从而影响到算法近似比。

H. Jiang 等人的 2-近似算法中, 步骤 1 在处理 1-Type-1 串时由于使用了贪婪策略, 会导致部分字符串错放而破坏最优解中的公共邻接。如图 7 所示, 将缺失基因 7 插入 S 中, 由于 7 为 1-Type-1 类型串, 应使用贪婪策略将 7 插入到 \boxed{m} 和 $\boxed{k, 7}$ 之间, 然而 7 实际并不是 1-Type-1 类型串, 而是 1-Type-2 类型串。这是因为将 7 插入到 \boxed{m} 和 $\boxed{k, 7}$ 之间会产生 $\{m7\}$ 和 $\{7k\}$ 两个邻接, 但 2-近似算法并没有考虑 $\{7k\}$ 并不是产生的新公共邻接, 而是原有邻接。事实证明将 7 插入到 \boxed{m} 和 $\boxed{k, 7}$ 之间并不是最优解, 所以邻接冗余的特殊情况后续需要处理。

基因序列 $G = \langle 1, 7, \dots, m, 7, k \rangle$

基因序列 $S = \langle \boxed{1} \dots \boxed{m} \cdot \boxed{k, 7} \cdot \rangle$

图 7 特殊情况举例

序列中存在连续长缺失基因串, 在完成 1-Type-1 类型串的插入后, 假设存在可构成 n -Type-1 类型串的缺失基因串 $a_1 a_2 \dots a_n$, 如若按照 2-近似算法中分别处理, 则该缺失基因串中 1-Type-1 类型串被破坏, 有以下 2 种情况:

(1) 最多有 n 个 1-Type-2 类型串, 产生 n 个邻接;

(2) 最少有 2 个 1-Type-2 类型串, 剩余为 n -Type-3 类型串, 产生 2 个邻接。

产生邻接数 k 为 $2 \leq k \leq n$, 若将其合并后插入基因序列中, 则会产生 $n+1$ 个邻接。以上证明合并插入 n -Type-1 类型串具有更优效果。

在处理 n -Type-2 类型串时,假设存在可构成 n -Type-2 类型串的缺失基因串 $b_1b_2\cdots b_n$,如若按照 2-近似算法分别处理,则该缺失基因串中 n -Type-2 类型串被破坏,有以下 2 种情况:

(1)最多有 2 个 1-Type-2 类型串,同时 $b_i(2 \leq i \leq n-1)$ 均可与其相邻缺失基因 b_{i-1} 或 $b_{i+1}(2 \leq i \leq n-1)$ 构成内部邻接,产生 n 个邻接;

(2)最少有 1 个 1-Type-2 类型串,同时剩余 $n-1$ 个缺失基因均为 n -Type-3 类型串,产生 1 个邻接;产生邻接数 k 为 $1 \leq k \leq n$,如若将其合并后插入基因序列中,会产生 n 个邻接。以上证明合并插入 n -Type-2 类型串具有更优效果。

对于 2.57-近似算法,虽然 n -Type-3 类型串会产生 $n-1$ 个邻接,但是如果不对其进行处理,任其插入未锁定的 slot,存在破坏已有邻接的可能,其中最多

会破坏 2 个邻接, n -Type-3 类型串产生邻接数 k 为 $n-3 \leq k \leq n-1$,明显降低最后的填充效率,所以对其进行插入处理十分必要。

k -Mer 算法固定了基因子串的长度,2-近似算法就是一个特例,其固定长度为 2 的公共邻接数目作为算法性能参考依据,以上也证明了此时并不是最优算法,进而说明了基因填充过程中限制基因子串长度存在影响近似性能比的可能。

4 总结与展望

重点介绍了基于 contig 的单面重复基因组片段填充问题的研究现状。对该些算法在近似性能比等方面做出了详细的对比(见表 1),直观地看出各个算法现存在的一定不足,说明此类算法仍有改进空间,同时提出了该类问题的改进思路。

表 1 三种算法分析比较

算法	核心技术	衡量依据	近似比	时间复杂度	改进思路
2-近似算法	最大匹配、贪心算法	公共邻接数目	2	$O(n^{2.5})$	考虑不同长度的缺失基因的插入情况
2.57-近似算法	最大匹配	公共邻接数目	2.57	$O(n^{2.5})$	考虑 Type-3 类型串的插入情况
k -Mer 算法	动态规划、图着色	公共 k -mers 数目	多项式时间可解	$2^{O(\ell)} \bullet n^{O(\ell)}$	考虑插入缺失串的特殊性,对缺失基因进行分类

4.1 面临的挑战

(1)现有算法依赖于公共邻接数目,邻接的定义没有考虑基因序列不存在逆序关系的情况,因此存在两个基因序列的邻接均为公共邻接但二者完全不相似的情况,不利于后续对算法近似比的研究。

(2)现有算法对度量依据的选择较少,并过度依赖于最大匹配算法,因此对此类问题的研究较为片面。

4.2 前景展望

针对目前单面重复基因组片段填充问题的研究工作,发现基因组填充问题有以下发展前景。

(1)目前 One-Sided-SF-max 问题的最佳性能近似比为 2,日后还需要进一步优化。

(2)现有算法均是在最大化邻接基础上考虑其近似比,基于最小断点数层面有待研究。

(3)双面基因组填充问题也被证为 NP-完全的,但没有提出近似比算法,此类算法可推广到双面基因组填充问题,有利于双面此问题的近似比优化。

参考文献:

[1] 周卫星,石海鹤.高通量测序中序列拼接算法的研究进展

[J]. 计算机科学,2019,46(5):36-43.

[2] 张 曦,樊晓桢,康继昌,等. DNA 质量筛选算法研究[J]. 计算机技术与发展,2015,25(7):41-44.

[3] 李瑞琳,尚秋明,韩鑫胤,等. 基于宏基因组组长片段的基因预测算法基准[J]. 计算机应用,2019,39(S1):143-149.

[4] BULTEAU L, CARRIERI A P, DONDIR. Fixed-parameter algorithms for scaffold filling[J]. Theoretical Computer Science, 2015, 568:72-83.

[5] 官登峰. 单倍体基因组序列组装方法研究[D]. 哈尔滨:哈尔滨工业大学,2020.

[6] GABOW H N. The weighted matching approach to maximum cardinality matching [J]. Fundamenta Informaticae, 2017, 154(1-4):109-130.

[7] LIU Nan, ZHU Daming, JIANG Haitao, et al. A 1.5-approximation algorithm for two-sided scaffold filling[J]. Algorithmica, 2016, 74(1):91-116.

[8] 朱 越. 基于 DNA 的连续优化算法[J]. 计算机工程与应用, 2011, 47(22):48-52.

[9] 刘国庆,曾艳丽,魏君锋,等. 基因序列拼接算法设计[J]. 计算机应用与软件, 2010, 27(5):24-26.

[10] 赵 倩. 公共邻接距离基因组片段填充问题研究[D]. 济南:山东大学,2013.

- [11] 王莉. 基因组片段填充问题的算法研究[D]. 济南: 山东大学, 2013.
- [12] MUÑOZ A, ZHENG Chunfang, ZHU Qian, et al. Scaffold filling, contig fusion and comparative gene order inference[J]. *BioMed Central*, 2010, 11(1): 304.
- [13] YANCOPOULOS S, ATTIE O, FRIEDBERG R. Efficient sorting of genomic permutations by translocation, inversion and block interchange[J]. *Bioinformatics*, 2005, 21(16): 3340–3346.
- [14] JIANG Haitao, ZHENG Chunfang, SANKOFF D, et al. Scaffold filling under the breakpoint and related distance[J]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2012, 9(4): 1220–1229.
- [15] ANGIBAUD S, FERTIN G, RUSU I, et al. On the approximability of comparing genomes with duplicates[C]//2nd international workshop on algorithms and computation. Dhaka; Springer, 2008: 34–45.
- [16] 柳楠. 基因组片段填充问题的算法研究[D]. 济南: 山东大学, 2013.
- [17] LIU N, JIANG H T, ZHU D M, et al. An improved approximation algorithm for scaffold filling to maximize the common adjacencies[J]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2013, 10(4): 905–913.
- [18] JIANG H T, MA J J, LUAN J F, et al. Approximation and nonapproximability for the one-sided scaffold filling problem[C]//21st international conference on computing and combinatorics. Beijing; Springer, 2015: 251–263.
- [19] MA J J, JIANG H T. Notes on the 6/5-approximation algorithm for one-sided scaffold filling[C]//10th international workshop on frontiers in algorithmics. Qingdao; Springer, 2016: 145–157.
- [20] ZHU B H. Genomic scaffold filling: a progress report[C]//10th international workshop on frontiers in algorithmics. Qingdao; Springer, 2016: 8–16.
- [21] JIANG H T, FAN C L, YANG B T, et al. Genomic scaffold filling revisited[C]//27th annual symposium on combinatorial pattern matching. Dagstuhl; Schloss Dagstuhl – Leibniz – Zentrum für Informatik GmbH, 2016: 1–13.
- [22] JIANG H T, QINGGE L T, ZHU D M, et al. A 2-approximation algorithm for the contig-based genomic scaffold filling problem[J]. *Journal of Bioinformatics and Computational Biology*, 2018, 16(6): 1850022.
- [23] BULTEAU L, FERTIN G, KOMUSIEWICZ C. Beyond adjacency maximization: scaffold filling for new string distances[C]//28th annual symposium on combinatorial pattern matching. Dagstuhl; Schloss Dagstuhl – Leibniz – Zentrum für Informatik GmbH, 2017.
- [24] TAN Guanlan, FENG Qilong, MENG Xiangzhong, et al. A new approximation algorithm for contig-based genomic scaffold filling[J]. *Theoretical Computer Science*, 2021, 853: 7–15.
- [25] LIU N, ZOU P, ZHU B H. A polynomial time solution for permutation scaffold filling[C]//10th annual international conference on combinatorial optimization and applications. Hong Kong, China; Springer, 2016: 782–789.